

Sommersemester 2015

Assignment on Geometric Data Structures for Computer Graphics - Sheet 5

Due Date 24. 06. 2015

Exercise 1 (BSP-Konstruktion, 6 Credits)

Geben Sie ein Beispiel für eine Menge von Liniensegmenten in der Ebene an, bei der die *Greedy*-Methode zur Konstruktion einer Auto-Partition (*greedy* bedeutet hier, dass die Splitting-Linie so gewählt wird, dass die Anzahl der Schnitte minimal ist) einen BSP-Tree von quadratischer Größe erzeugt.

Exercise 2 (BSP- vs kd-Tree, 3+5 Credits)

Eigentlich sind kd-Trees lediglich ein Spezialfall von BSP-Trees; Die Splitting-Linien dürfen lediglich nicht frei gewählt werden, sondern müssen immer horizontal oder vertikal verlaufen.

- a) Geben Sie mindestens 3 Vor- und 3 Nachteile von BSP-Trees gegenüber kd-Trees an.
- b) Für je zwei sich nicht schneidende Liniensegmente in der Ebene kann man trivialerweise einen BSP-Tree der Größe 2 erzeugen. Für kd-Trees gilt dies nicht: Zeigen Sie, dass es keine Konstante c gibt, so dass für jedes Paar sich nicht schneidender Liniensegmente ein kd-Tree von maximaler Größe c existiert.

Hinweis zu kd-Trees für Liniensegmente: Hier wird so lange unterteilt und zerschnitten, bis sich maximal ein einzelnes Liniensegment in jedem Blatt des kd-Trees befindet.

Exercise 3 (Painters Algorithmus, 6 Credits)

Beweisen Sie, dass der Painters-Algorithmus korrekt ist. Zeigen Sie dazu, dass wenn (ein Teil von) Objekt A vor (Teilen von) Objekt B scan-konvertiert wird, so kann Objekt A nicht vor Objekt B liegen.